

Docket No. AUS920030439US1

CLAIMS:

What is claimed is:

- 5 1. A method of enhancing priority boosting of scheduled threads comprising the steps of:

10 determining by a second thread being executed on a second CPU whether to wait for a lock on a shared resource held by a first thread that is scheduled to be executed by a first CPU, the second thread having a higher priority than the first thread;

15 boosting the priority of the first thread by passing the higher priority of the second thread to the first thread; and

20 enhancing the priority boosting of the first thread by rescheduling the first thread to run on the second CPU.

- 25 2. The method of Claim 1 wherein when the priority boosting of the first thread is enhanced, the first thread becomes the next thread to be dispatched for execution.

- 30 3. The method of Claim 2 wherein after the priority boosting of the first thread has been enhanced, the second thread relinquishes the second CPU by going to sleep awaiting the release of the lock by the first thread.

Docket No. AUS920030439US1

4. The method of Claim 3 wherein after the first thread has released the lock, the second thread is awakened and rescheduled for execution by the second CPU.
- 5 5. The method of Claim 4 wherein after being rescheduled for execution by the second CPU, the second thread is likely the next thread to be dispatched for execution.
6. A computer program product on a computer readable
10 medium for enhancing priority boosting of scheduled threads comprising:

code means for determining by a second thread being
executed on a second CPU whether to wait for a lock on
15 a shared resource held by a first thread that is scheduled to be executed by a first CPU, the second thread having a higher priority than the first thread;

code means for boosting the priority of the first
20 thread by passing the higher priority of the second thread to the first thread; and

code means for enhancing the priority boosting of the
first thread by rescheduling the first thread to run on
25 the second CPU.
7. The computer program product of Claim 6 wherein when the priority boosting of the first thread is enhanced, the first thread becomes the next thread to be
30 dispatched for execution.

Docket No. AUS920030439US1

8. The computer program product of Claim 7 wherein after the priority boosting of the first thread has been enhanced, the second thread relinquishes the second CPU by going to sleep awaiting the release of the lock by the first thread.
9. The computer program product of Claim 8 wherein after the first thread has released the lock, the second thread is awakened and rescheduled for execution by the second CPU.
10. The computer program product of Claim 9 wherein after being rescheduled for execution by the second CPU, the second thread is likely the next thread to be dispatched for execution.
11. An apparatus for enhancing priority boosting of scheduled threads comprising:
- means for determining by a second thread being executed on a second CPU whether to wait for a lock on a shared resource held by a first thread that is scheduled to be executed by a first CPU, the second thread having a higher priority than the first thread;
- means for boosting the priority of the first thread by passing the higher priority of the second thread to the first thread; and
- means for enhancing the priority boosting of the first thread by rescheduling the first thread to run on the second CPU.

Docket No. AUS920030439US1

12. The apparatus of Claim 11 wherein when the priority
boosting of the first thread is enhanced, the first
thread becomes the next thread to be dispatched for
5 execution.

13. The apparatus of Claim 12 wherein after the priority
boosting of the first thread has been enhanced, the
second thread relinquishes the second CPU by going to
10 sleep awaiting the release of the lock by the first
thread.

14. The apparatus of Claim 13 wherein after the first
thread has released the lock, the second thread is
15 awakened and rescheduled for execution by the second
CPU.

15. The apparatus of Claim 14 wherein after being
rescheduled for execution by the second CPU, the second
20 thread is likely the next thread to be dispatched for
execution.

16. A system for enhancing priority boosting of scheduled
threads comprising:
25

at least one storage device for storing code data; and

at least two CPUs, one of the at least two CPUs for
processing the code data to determine by a second
30 thread being executed on the at least one CPU whether
to wait for a lock on a shared resource held by a first
thread that is scheduled to be executed by the other

Docket No. AUS920030439US1

5 CPU, the second thread having a higher priority than the first thread, to boost the priority of the first thread by passing the higher priority of the second thread to the first thread, and to enhance the priority boosting of the first thread by rescheduling the first thread to run on the second CPU.

10 17. The system of Claim 16 wherein when the priority boosting of the first thread is enhanced, the first thread becomes the next thread to be dispatched for execution.

15 18. The system of Claim 17 wherein after the priority boosting of the first thread has been enhanced, the second thread relinquishes the second CPU by going to sleep awaiting the release of the lock by the first thread.

20 19. The system of Claim 18 wherein after the first thread has released the lock, the second thread is awakened and rescheduled for execution by the second CPU.

25 20. The system of Claim 19 wherein after being rescheduled for execution by the second CPU, the second thread is likely the next thread to be dispatched for execution.